CARDIOCOM®

Cardiocom, LLC

CC FM TLS/SRTP

FIPS 140-2 Cryptographic Module Non-Proprietary Security Policy

Version: 1.6

Date: February 11, 2016

CARDIOCOM®

Table of Contents

List of Tables

List of Figures

1 Introduction

This document defines the Security Policy for the Cardiocom CC FM TLS/SRTP module, hereafter denoted as the "Module". The Module is designed to support TLS and SRTP/RTCP usage scenarios. This enables support for secure communication using standard internet technologies. The Module meets FIPS 140-2 overall Level 1 requirements.

Table 1 – Cryptographic Module Configurations

SW Version	Operating Environment
1.0.2	Windows Server 2008 R2 (x64) Android 4.0.4

The Module is intended for use by US Federal agencies and other markets that require FIPS 140-2 validated software library. The Module is a multi-chip standalone embodiment; the cryptographic boundary is a single DLL (Windows) or SO (Android) file.

The FIPS 140-2 security levels for the Module are as follows:

Table 2 – Security Level of Security Requirements

Security Requirement	Security Level
Cryptographic Module Specification	1
Cryptographic Module Ports and Interfaces	1
Roles, Services, and Authentication	1
Finite State Model	1
Physical Security	N/A
Operational Environment	1
Cryptographic Key Management	1
EMI/EMC	1
Self-Tests	1
Design Assurance	1
Mitigation of Other Attacks	N/A

The Module implementation is compliant with:

- RFC 5246 - Transport Layer Security (TLS) Protocol Version 1.2
- RFC 3711 – The Secure Real-time Protocol (SRTP)

1.1 Cryptographic Boundary

The physical boundary of the module is the general purpose computer or device. The physical ports are that of a general purpose computer/device.

The logical boundary is a single DLL (Windows) [Cardiocom_Fips_Module.dll] or SO (Android) [libccfm.so] file.

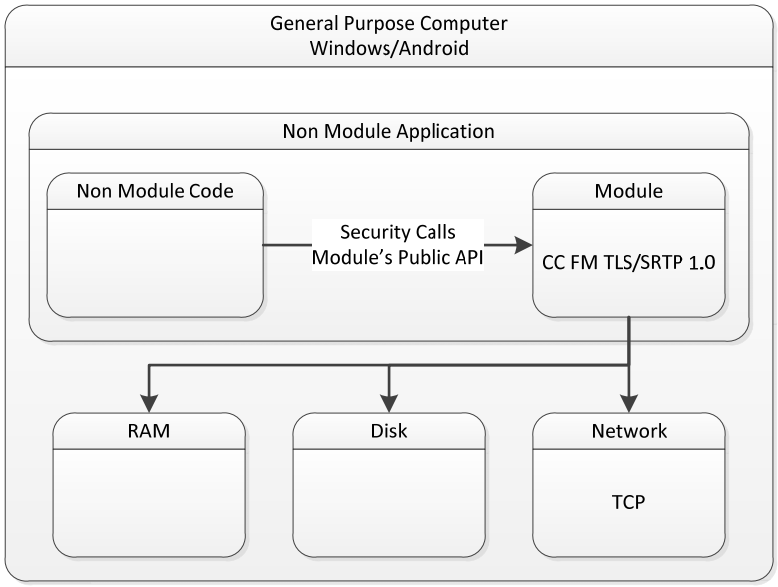

Figure 1 – Module Block Diagram

Table 3 – Ports and Interfaces

Port	Description	Logical Interface Type
API	Enumerated in services section of this document.	Control in | Data in | Data out | Status out
Camera	Entropy source for Android	Data in
TCP	TLS is tied to the TCP Transport Protocol	Control in | Data in | Data out | Status out

1.2 Mode of Operation

The module supports two (2) explicitly selected modes of operation. By default, the module uses the FIPS approved SRTP KDF. It's also possible to call the function CC_FM_SRTP_Set_KDF_Mode and choose an SRTP KDF that's associated with the shorter key sizes and defined in RFC 3711. This feature is included within this module to balance compliance with compatibility. This function modifies an internal variable that is used on subsequent SRTP calls. The FIPS Approved mode of operation uses only FIPS Approved algorithms. Use of the non-Approved TLS/SRTP implementation or non-approved key sizes constitutes the entry into the non-Approved mode. To verify that a module is in the Approved mode of operation, the operator must ensure that the TLS/SRTP algorithm being utilized is the FIPS Approved version. Connections established utilizing the non-Approved version is not operating in Approved mode (see Table 4 below).

Non-Approved Modes Key Sizes (TLS):

Table 4: Non-Approved Algorithms

SRTP-KDF (non-compliant)	Non-CAVP tested KDF

2 Cryptographic Functionality

The Module implements the FIPS Approved and Non-Approved but Allowed cryptographic functions listed in the tables below.

Table 5 – Approved and CAVP Validated Cryptographic Functions

Algorithm	Description	Cert #
AES	[FIPS 197, SP 800-38A] Functions: Encryption, Decryption Modes (Key Sizes): ECB (128 bits, 256 bits) , CBC (128 bits), CTR (256 bits)	3349
DRBG	[SP 800-90A] Functions: CTR DRBG Security Strengths: 256 bits	794, 795
HMAC	[FIPS 198-1] Functions: Generation, Verification SHA sizes: SHA-1, SHA-256	2132
KDF, Existing Application-Specific	[SP 800-135] Functions: TLS 1.2 KDF, SRTP KDF	494, 495 (CVL)
RSA	[FIPS 186-4, PKCS1.5 Functions: Signature Verification Key sizes: 1024, 2048 bits * Signature Generation has been tested, but is not used by the module	1716
SHA	[FIPS 180-4] Functions: Digital Signature Generation, Digital Signature Verification, non-Digital Signature Applications SHA sizes: SHA-1, SHA-256, SHA-512	2776

Table 6 – Non-Approved but Allowed Cryptographic Functions

Algorithm	Description
Non-SP 800-56A Compliant DHE	[IG D.8] Diffie-Hellman (key agreement; key establishment methodology provides 112 bits of security strength) using 2048 bit keys.
NDRNG	[Annex C] Non-Deterministic RNG pools from different entropy sources (Microsoft: CryptGenRandom, Android: Camera); minimum of 1.6 kbs per access. The NDRNG output is used to seed the FIPS Approved DRBG.

Table 7 – Protocols Allowed in FIPS Mode

Protocol	Description
SRTP	[IG D.8 and SP 800-135] Cipher Suites: AES_256_SHA1_80 *(This protocol has not been reviewed or tested by the CAVP and CMVP)*
TLS v1.2	[IG D.8 and SP 800-135] Cipher Suites: TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 *(This protocol has not been reviewed or tested by the CAVP and CMVP)*

2.1 Critical Security Parameters

All CSPs used by the Module are described in this section. All usage of these CSPs by the Module (including all CSP lifecycle states) is described in the services detailed in Section 4.

Table 8 – Critical Security Parameters (CSPs)

CSP	Description / Usage
TLS Server PEM Formatted Private Key	TLS Server private key used to establish secure connection. Key lengths of 1024 (non-approved) and 2048 (approved) - RSA
TLS DHE Private Components	Supporting DHE key agreement. (Key length of 2048)
TLS Pre Master Secret	TLS pre-master component used to establish secure connection. (384 bits via SP800-90A DRBG)
TLS Session Key	TLS key used to secure current session. (384 bits via TLS KDF)
SRTP/RTCP Master Secret	SRTP master secret in support of secure connection. (368 bits via SP800-90A DRBG)
SRTP/RTCP Session Key	SRTP key used to secure current session (256 bits via SRTP KDF).
DRBG V and Key	DRBG Internal State Values (IV= 32 bits and Key = 256 bits)

2.2 Public Keys

Table 9 – Public Keys

Key	Description / Usage
TLS Server PEM Formatted CA	TLS Server root cert used to establish server cert. (RSA 1024 & 2048)
TLS Server PEM Formatted Server Cert	TLS Server cert derived from the CA cert. (RSA 1024 & 2048)
TLS DHE Public Components	Supporting DHE key agreement. (Key length of 2048)
SW Integrity Verification Key	Verifies the integrity of the SW. (RSA 2048 with SHA-512)

3 Roles, Authentication and Services

3.1 Assumption of Roles

The module supports two (2) distinct operator roles, User and Cryptographic Officer (CO). Both roles are able to call all public functions on the module. But their roles describe access to different types of CSP values.

- User – Perform client activities (TLS Client, SRTP/RTCP).
- CO – Access to all User role parameter values. In addition, the CO role has access to private keys (TLS Server).

The Module does not support a maintenance role or bypass capability. The Module does not support concurrent operators.

3.2 Services and CSP Access Rights

All services implemented by the Module are listed in the table below. Each service description also describes all usage of CSPs by the service. The module provides identical services to both the approved and non-approved modes. In the non-Approved mode of operation, the services listed below utilize the non-Approved algorithms from Table 4 above in lieu of the module's Approved algorithms.

Each function in Table 10 –Services, has different modes of access. The following table defines these access codes:

- G = Generate: The module generates the CSP.
- R = Read: The module reads the CSP. The read access is typically performed before the module uses the CSP.
- E = Execute: The module executes using the CSP.
- W = Write: The module writes the CSP. The write access is typically performed after a CSP is imported into the module, when the module generates a CSP, or when the module overwrites an existing CSP.
- Z = Zeroize: The module zeroizes the CSP.

Table 10 –Services and CSP Access Rights

Service	Description	CO	User
Show Status	A status code is returned from each function. An error description can be obtained by calling CC_FM_ErrorCodeToString. CSP usage: None	x	x
Perform Self-Tests	Self-tests can be performed on demand by reloaded the library or by calling CC_FM_RunFipsTests. CSP usage: None	x	x
Zeroize	This topic can be broken out by each major server: • TLS will read incoming CSP values from disk and zeroize this memory inside the module with appropriate API: `CC_FM_TLS_Client_ContextRelease (Z), CC_FM_TLS_Server_ConnectionClose (Z), CC_FM_TLS_Server_ContextRelease (Z).` • SRTP/RTCP requires that a master password be generated. There's a corresponding module API CC_FM_SRTP_ReleasePassword (Z) for securely zeroizing this value. CSP Usage: All CSPs are zeroized.	x	x
Library Global	Support functions. • CC_FM_ErrorCodeToString • CC_FM_RunFipsTests • CC_FM_TestResultsRelease • CC_FM_RNG_NonDeterministic_GetBytes CSP usage: None	x	
TLS Client	Functions that facilitate performing the TLS protocol from a client perspective. • CC_FM_TLS_Client_ContextInit (R) • CC_FM_TLS_Client_ContextInfo • CC_FM_TLS_Client_Connect (E) • CC_FM_TLS_Client_ConnectionInfo • CC_FM_TLS_Client_Send (E) • CC_FM_TLS_Client_Receive (E) • CC_FM_TLS_Client_ConnectionClose • C_FM_TLS_Client_ContextRelease (Z) CSP Usage: (G, E, R & W) TLS DHE Key, TLS Pre Master Secret, TLS Session Key	x	x

Service	Description	CO	User
TLS Server	Functions that facilitate performing the TLS Server protocol from the server perspective. • CC_FM_TLS_Server_ContextInit (R) • CC_FM_TLS_Server_ContextInfo • CC_FM_TLS_Server_ConnectionAccept (R) • CC_FM_TLS_Server_ConnectionInfo • CC_FM_TLS_Server_Send (E) • CC_FM_TLS_Server_Receive (E) • CC_FM_TLS_Server_ConnectionClose (Z) • CC_FM_TLS_Server_ContextRelease (Z) CSP Usage: TLS Server PEM Formatted Private Key (R), TLS DHE Key (G, E), TLS Pre Master Secret (G, E), TLS Session Key (G, E)	x	
SRTP/RTCP	Functions that facilitate SRTP/RTCP communication from both a client and server perspective. • CC_FM_SRTP_Session_Init (R) • CC_FM_SRTP_AllocatePassword (G) • CC_FM_SRTP_ReleasePassword (Z) • CC_FM_SRTP_StreamAdd (R) • CC_FM_SRTP_Protect (E) • CC_FM_SRTP_Unprotect (E) • CC_FM_SRTP_Protect_RTCP (E) • CC_FM_SRTP_Unprotect_RTCP (E) • CC_FM_SRTP_StreamRemove (Z) • CC_FM_SRTP_Session_Release (Z) • CC_FM_SRTP_Set_KDF_Mode (E) CSP Usage: (G, R, E, W) SRTP/RTCP Master Secret, SRTP/RTCP Session Key	x	x

4 Self-tests

Each time the Module is powered up, it tests that the cryptographic algorithms still operate correctly. The power up self-tests are available on demand by calling the Perform Self-Tests service.

On power up or reset, the Module performs self-tests described in Table 11 below. All KATs must be completed successfully prior to any other use of cryptography by the Module. If one of the KATs fails, the Module enters the globally defined FIPS error state. At the beginning of each public function, this state is checked. If the module is in error, the function exits with an appropriate error code.

Table 11 – Power Up Self-tests

Test Target	Description
Software Integrity	RSA 2048 SHA512 signature verification. (RSACert. #1716 and SHA Cert. #2776)
AES	KATs: Encryption, Decryption (Cert. #3349) Modes: ECB, CBC, CTR Key sizes: 128, 256
DRBG	KATs: CTR DRBG (Certs. #794 and #795)
HMAC	KATs: Generation, Verification (Cert. #2132) SHA sizes: SHA-1, SHA-256
RSA	KATs: Signature Verification (Cert. #1716) Key sizes: 2048 bits
SHA	KATs: SHA-1, SHA-256, SHA-512 (Cert. #2776)
TLS-KDF, SRTP-KDF	KATs: TLS KDF, SRTP KDF (Certs. #494 and #495)

Table 12 – Conditional Self-tests

Test Target	Description
NDRNG	NDRNG Continuous Test performed when a random value is requested from the NDRNG.
DRBG	DRBG Continuous Test performed when a random value is requested from the DRBG.
DRBG Health Checks	Performed conditionally per SP 800-90 Section 11.3.

5 Operational Environment

The module was tested on the following operating environments:

- Windows Server 2008 R2 (x64)
- Android 4.0.4

Each operational environment listed limits access to the module by a single operator at a time. In this case, the calling application is considered the operator, and as such, the module is being utilized by a single operator at a time.

6 Mitigation of Other Attacks Policy

The module does not mitigate any attacks outside the scope of FIPS 140-2.

7 Security Rules and Guidance

The Module design corresponds to the Module security rules. This section documents the security rules enforced by the cryptographic module to implement the security requirements of this FIPS 140-2 Level 1 module.

1. The operator shall be capable of commanding the module to perform the power up self-tests by cycling power or resetting the module.
2. Power up self-tests do not require any operator action.
3. Data output shall be inhibited during key generation, self-tests, zeroization, and error states.
4. Status information does not contain CSPs or sensitive data that if misused could lead to a compromise of the module.
5. There are no restrictions on which keys or CSPs are zeroized by the zeroization service.
6. Operator must remain in control of the module during the zeroization process.
7. The module does not support concurrent operators.
8. The module does not support a maintenance interface or role.
9. The module does not support manual key entry.
10. The module does not have any external input/output devices used for entry/output of data.
11. The module does not output intermediate key values.

8 References and Definitions

The following standards are referred to in this Security Policy.

Table 13 – References

Abbreviation	Full Specification Name
[FIPS140-2]	*Security Requirements for Cryptographic Modules*, May 25, 2001
[SP800-131A]	*Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths*, January 2011

Table 14 – Acronyms and Definitions

Acronym	Definition
API	Application Program Interface
CC_FM	Cardiocom FIPS Module
DLL	Dynamic Link Library (Windows)
KDF	Key Derivation Function
SO	Shared Object (Android)
RTCP	RTP Control Protocol
SRTP	Secure Real-Time Transport Protocol
TCP	Transport Control Protocol
TLS	Transport Layer Security